

KODE MODUL

SWR.DEV.100.(1).A



**SEKOLAH MENENGAH KEJURUAN
BIDANG KEAHLIAN TEKNOLOGI INFORMASI DAN KOMUNIKASI
PROGRAM KEAHLIAN REKAYASA PERANGKAT LUNAK**

MENYIAPKAN DAN MELAKUKAN SURVEY UNTUK MENENTUKAN KEBUTUHAN DATA



**BAGIAN PROYEK PENGEMBANGAN KURIKULUM
DIREKTORAT PENDIDIKAN MENENGAH KEJURUAN
DIREKTORAT JENDERAL PENDIDIKAN DASAR DAN MENENGAH
DEPARTEMEN PENDIDIKAN NASIONAL
2004**

KODE MODUL

SWR.DEV.100.(1).A



SEKOLAH MENENGAH KEJURUAN
BIDANG KEAHLIAN TEKNOLOGI INFORMASI DAN KOMUNIKASI
PROGRAM KEAHLIAN REKAYASA PERANGKAT LUNAK

Menyiapkan dan Melakukan Survey untuk Menentukan Kebutuhan Data



BAGIAN PROYEK PENGEMBANGAN KURIKULUM
DIREKTORAT PENDIDIKAN MENENGAH KEJURUAN
DIREKTORAT JENDERAL PENDIDIKAN DASAR DAN MENENGAH
DEPARTEMEN PENDIDIKAN NASIONAL

2004

KODE MODUL

SWR.DEV.100.(1).A



SEKOLAH MENENGAH KEJURUAN
BIDANG KEAHLIAN TEKNOLOGI INFORMASI DAN KOMUNIKASI
PROGRAM KEAHLIAN REKAYASA PERANGKAT LUNAK

Menyiapkan dan Melakukan Survey untuk Menentukan Kebutuhan Data



BAGIAN PROYEK PENGEMBANGAN KURIKULUM
DIREKTORAT PENDIDIKAN MENENGAH KEJURUAN
DIREKTORAT JENDERAL PENDIDIKAN DASAR DAN MENENGAH
DEPARTEMEN PENDIDIKAN NASIONAL
2004

KODE MODUL

SWR.DEV.100.(1).A



SEKOLAH MENENGAH KEJURUAN
BIDANG KEAHLIAN TEKNOLOGI INFORMASI DAN KOMUNIKASI
PROGRAM KEAHLIAN REKAYASA PERANGKAT LUNAK

Menyiapkan dan Melakukan Survey untuk Menentukan Kebutuhan Data

PENYUSUN
TIM FAKULTAS TEKNIK
UNIVERSITAS NEGERI YOGYAKARTA

BAGIAN PROYEK PENGEMBANGAN KURIKULUM
DIREKTORAT PENDIDIKAN MENENGAH KEJURUAN
DIREKTORAT JENDERAL PENDIDIKAN DASAR DAN MENENGAH
DEPARTEMEN PENDIDIKAN NASIONAL
2005

KATA PENGANTAR

Modul dengan judul “Menyiapkan dan Melakukan Survey untuk Menentukan Kebutuhan Data” merupakan bahan ajar yang digunakan sebagai panduan praktikum peserta diklat Sekolah Menengah Kejuruan (SMK) untuk membentuk salah satu bagian dari kompetensi bidang keahlian Teknologi Informasi dan Komunikasi pada Program Keahlian Rekayasa Perangkat Lunak.

Modul ini menguraikan tentang cara atau proses pembuatan algoritma dan diagram alur pemrograman sebagai sarana untuk survey dan menentukan kebutuhan data. Pembuatan algoritma dan diagram alir pemrograman ini untuk berbagai macam kasus yang sering terjadi di lapangan. Kegiatan Belajar 1 membahas tentang konsep algoritma dan diagram alir pemrograman, Kegiatan Belajar 2 membahas tentang Algoritma dan Diagram Alir untuk Proses Pengulangan (Loop), dan Kegiatan Belajar 3 membahas tentang Operasi Pada Struktur Data Larik 1 Dimensi.

Modul ini terkait dengan modul-modul lain yang membahas tentang Mengoperasikan Sistem Operasi, Menginstall Software, dan Mengubah Konfigurasi software. Oleh karena itu, sebelum menggunakan modul ini peserta diklat diwajibkan telah mengambil modul-modul tersebut.

Yogyakarta, Desember 2004

Penyusun

Tim Fakultas Teknik

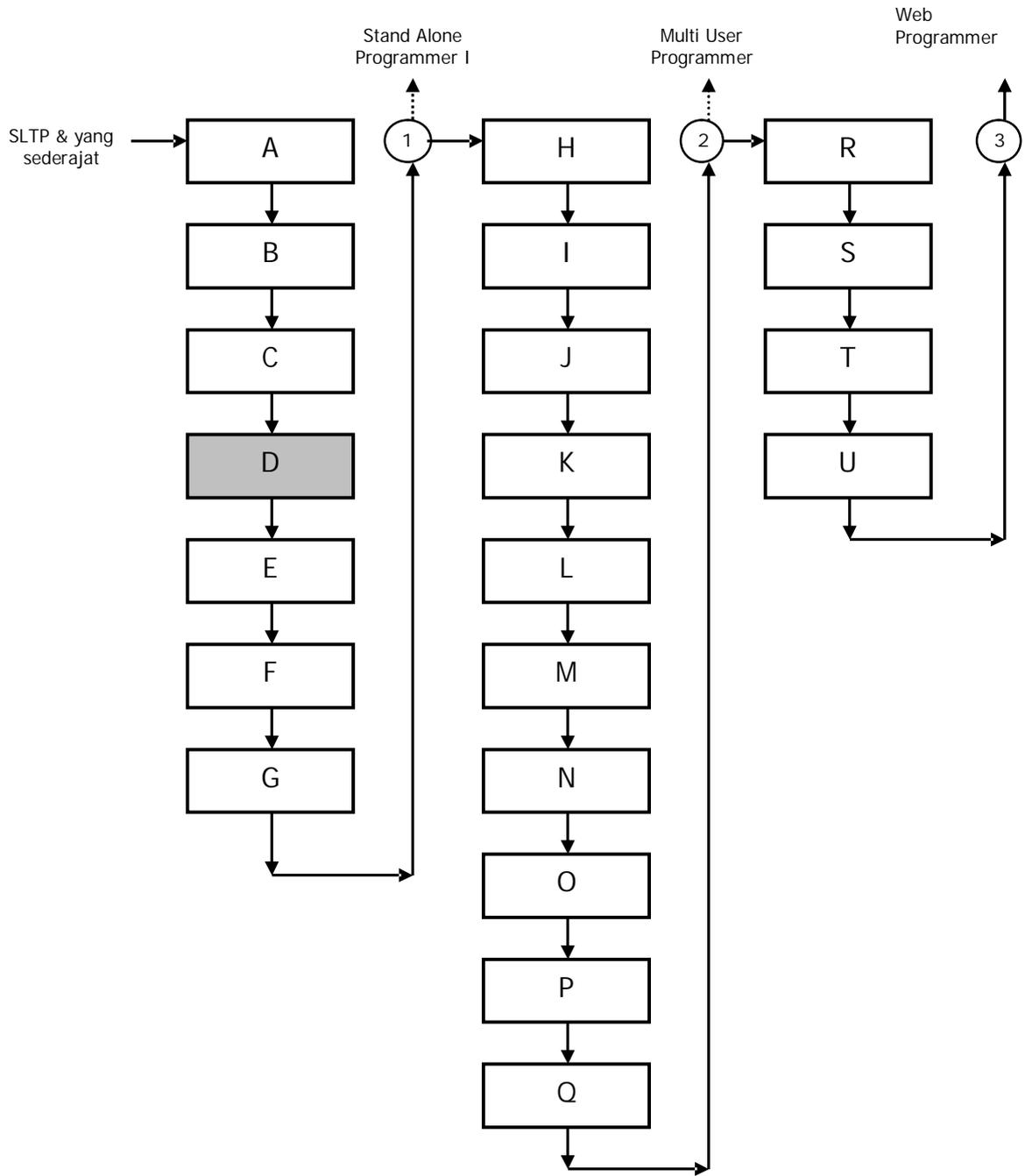
Universitas Negeri Yogyakarta

DAFTAR ISI MODUL

	Halaman
HALAMAN DEPAN	i
HALAMAN DALAM	ii
KATA PENGANTAR	iii
DAFTAR ISI MODUL	iv
PETA KEDUDUKAN MODUL	vi
PERISTILAHAN / GLOSSARY	viii
I. PENDAHULUAN	1
A. DESKRIPSI JUDUL	1
B. PRASYARAT	1
C. PETUNJUK PENGGUNAAN MODUL	2
1. Petunjuk Bagi Siswa	2
2. Peran Guru	3
D. TUJUAN AKHIR	3
E. KOMPETENSI	4
F. CEK KEMAMPUAN	6
II. PEMELAJARAN	7
A. RENCANA PEMELAJARAN SISWA	7
B. KEGIATAN BELAJAR	8
1. Kegiatan Belajar 1 : Konsep Algoritma dan Diagram Alir..	8
a. Tujuan Kegiatan Pemelajaran.....	8
b. Uraian Materi 1	8
c. Rangkuman 1.....	15
d. Tugas 1	16
e. Tes Formatif 1.....	16
f. Kunci Jawaban Formatif 1	16
g. Lembar Kerja 1	18

2. Kegiatan Belajar 2 : Algoritma dan Diagram Alir Untuk Proses	
Pengulangan	19
a. Tujuan Kegiatan Pemelajaran	19
b. Uraian Materi 2	19
c. Rangkuman 2.....	22
d. Tugas 2	22
e. Tes Formatif 2.....	22
f. Kunci Jawaban Formatif 2.....	23
g. Lembar Kerja 2	26
3. Kegiatan Belajar 3 : Operasi Pada Struktur Data Larik	
1 Dimensi.....	28
a. Tujuan Kegiatan Pemelajaran	28
b. Uraian Materi 3	28
c. Rangkuman 3.....	33
d. Tugas 3	33
e. Tes Formatif 3.....	33
f. Kunci Jawaban Formatif 3.....	34
g. Lembar Kerja 3	36
III. EVALUASI	38
A. PERTANYAAN	38
B. KUNCI JAWABAN EVALUASI	38
C. KRITERIA KELULUSAN	43
IV. PENUTUP	44
DAFTAR PUSTAKA	45

PETA KEDUDUKAN MODUL



Keterangan :

Kode	Kode Kompetensi	Kompetensi
A	SWR.OPR.200.(1).A	Mengoperasikan Sistem Operasi
B	SWR.OPR.100.(1).A	Menginstalasi software
C	SWR.MNT.100.(1).A	Mengubah konfigurasi software
D	SWR.DEV.100.(1).A	Menyiapkan dan melakukan survey untuk menentukan kebutuhan data
E	SWR.OPR.309.(1).A	Mengoperasikan software bahasa pemrograman level 1
F	DTA.OPR.115.(1).A	Konversi data level 1
G	SWR.DEV.500.(1).A	Menguji program level 1
H	HDW.OPR.103.(1).A	Mengoperasikan sistem operasi jaringan komputer berbasis teks
I	HDW.OPR.104.(1).A	Mengoperasikan sistem operasi jaringan komputer berbasis GUI
J	DTA.DEV.101.(3).A	Melakukan perancangan pengumpulan data
K	SWR.DEV.300.(2).A	Melakukan desain dan perancangan software
L	SWR.DEV.400.(2).A	Melakukan pengkodean program
M	DTA.MNT.101.(2).A	Melakukan back up data
N	DTA.MNT.102.(2).A	Melakukan restore data
O	SWR.OPR.303.(2).A	Mengoperasikan software aplikasi basis data
P	DTA.OPR.119.(2).A	Membuat query data
Q	SWR.DEV.500.(2).A	Menguji program
R	SWR.DEV.401.(2).A	Membangun interface dengan bahasa pemrograman berorientasi objek
S	SWR.DEV.402.(3).A	Melakukan pengkodean program
T	SWR.OPR.304.(3).A	Mengoperasikan bahasa pemrograman berbasis web
U	SWR.DEV.403.(2).A	Membangun program aplikasi remote data interaktif

PERISTILAHAN/GLOSSARY

- Algoritma : urutan langkah pekerjaan yang ditulis dalam notasi diskriptif yang logis untuk pencapaian suatu solusi atau untuk menuju ke suatu tujuan tertentu.
- Algoritma pemrograman : adalah urutan instruksi yang disusun sedemikian rupa sehingga mempunyai urutan nalar yang tepat untuk menyelesaikan suatu persoalan.
- Buble sort : Salah satu metoda atau teknik pengurutan data yang tidak efisien.
- Goto... : Instruksi pada bahasa pemrograman pascal untuk melakukan proses percabangan tak bersyarat.
- Loop : Kalang, iterasi atau pengulangan suatu proses.
- Mod : Salah satu operator aritmatika pada bahasa Pascal untuk mendapatkan sisa hasil bagi.
- Operator : Tanda yang digunakan dalam proses operasi aritmatik misalnya operasi perkalian dengan '*', penjumlahan dengan '+', pengurangan dengan '-', dll.
- Pseudocode : cuplikan dari kode-kode program yang dianggap penting.
- RPL : singkatan dari Rekayasa Perangkat Lunak.
- User : Pengguna, pemakai yakni seseorang yang menggunakan, mengoperasikan program kita.
- Variabel : Peubah, ubahan

BAB I

PENDAHULUAN

A. DESKRIPSI JUDUL

Menyiapkan dan Melakukan Survey untuk Menentukan Kebutuhan Data merupakan modul teori dan atau praktikum yang membahas tentang langkah-langkah awal dalam melakukan pemrograman komputer bagi pemula.

Modul ini terdiri dari 3 (tiga) kegiatan belajar, yaitu Konsep Algoritma dan Diagram Alir, Algoritma dan Diagram Alir Untuk Proses Pengulangan, Operasi Pada Struktur Data Larik Satu Dimensi. Dengan menguasai modul ini diharapkan peserta diklat mampu menggunakan algoritma pemrograman, mengidentifikasi kebutuhan data, dan membuat diagram alir data dan program untuk berbagai kasus yang sering terjadi di lapangan.

B. PRASYARAT

Kemampuan awal yang dipersyaratkan untuk mempelajari modul ini adalah :

- 1) Peserta diklat telah lulus modul / materi diklat Mengoperasikan Sistem Operasi berbasis GUI.
- 2) Peserta diklat telah lulus modul / materi Menginstalasi Software.
- 3) Peserta diklat telah lulus modul / materi Mengubah konfigurasi software.
- 4) Peserta diklat mampu mengoperasikan piranti-piranti periferan (misal: printer, monitor, floppy disk drive, CDROM drive).
- 5) Peserta diklat mampu menggunakan algoritma pemrograman dasar.
- 6) Peserta diklat sudah memahami tentang aljabar boolean dan fisika.

C. PETUNJUK PENGGUNAAN MODUL

1. Petunjuk Bagi Siswa

Siswa diharapkan mampu berperan aktif dan berinteraksi dengan sumber belajar yang mendukungnya, karena itu harus diperhatikan beberapa hal sebagai berikut :

a. Langkah – langkah Belajar

Modul ini berisi materi mengenai kompetensi menyiapkan dan melakukan survey untuk menentukan kebutuhan data, oleh sebab itu perlu diperhatikan beberapa hal agar peserta diklat lebih berkompeten dan profesional, yaitu :

- 1) Apa yang harus diketahui tentang sistem operasi berbasis GUI?
- 2) Apa yang harus dikerjakan dalam proses menginstall suatu software aplikasi?
- 3) Apa yang harus dikerjakan pertama kali dalam persiapan dan survey untuk menentukan kebutuhan data?
- 4) Bagaimana mengetahui bahwa kita telah berhasil atau belum dalam menyiapkan dan melakukan survey untuk menentukan kebutuhan data?
- 5) Apa yang harus dilakukan jika kita belum berhasil?
- 6) Apa yang harus dilakukan jika kita telah berhasil?
- 7) Apakah kita sudah sesuai aturan dalam menyiapkan dan melakukan survey untuk menentukan kebutuhan data?

b. Perlengkapan yang Harus Dipersiapkan

Untuk menunjang keselamatan kerja dan kelancaran tugas yang akan Anda lakukan, maka persiapkanlah seluruh perlengkapan yang diperlukan sesuai dengan jenis tugas pada masing-masing kegiatan pembelajaran.

c. Hasil Pelatihan

Anda akan mampu melakukan tugas/pekerjaan menyiapkan dan melakukan survey untuk menentukan kebutuhan data yang sesuai dengan program aplikasi yang akan kita bangun.

2. Peran Guru

Guru yang akan mengajarkan modul ini hendaknya mempersiapkan diri sebaik-baiknya yaitu mencakup aspek strategi pembelajaran, penguasaan materi, pemilihan metode, alat bantu media pembelajaran, dan perangkat evaluasinya.

D. TUJUAN AKHIR

1. Peserta diklat mampu menggunakan fungsi dan operator algoritma pemrograman.
2. Peserta diklat mampu mengidentifikasi dan menentukan kebutuhan data.
3. Peserta diklat mampu menyusun hasil survey di lapangan menjadi suatu acuan pembuatan program.
4. Peserta diklat mampu mempraktekkan pembuatan alur data dan diagram alir program.

E. KOMPETENSI

SUB KOMPETENSI	KRITERIA KINERJA	LINGKUP BELAJAR	MATERI POKOK PEMELAJARAN		
			SIKAP	PENGETAHUAN	KETRAMPILAN
1	2	3	4	5	6
1. Memahami prosedur survey penentuan kebutuhan data	<ul style="list-style-type: none"> § Prosedur survey untuk menentukan kebutuhan dapat dijelaskan sesuai jenis pekerjaan yang akan dilakukan § Prosedur operasi fungsi dan operator pada algoritma pemrograman 	<ul style="list-style-type: none"> § Konsep algoritma pemrograman § Diagram alir data 	<ul style="list-style-type: none"> § Men-rangkan konsep algoritma pemrograman 	<ul style="list-style-type: none"> § Menunjuk-kan fungsi algoritma pemrograman 	<ul style="list-style-type: none"> § Mengguna-kan fungsi dan operator algoritma pemrograman
2. Mempersiapkan pokok-pokok kebutuhan data	<ul style="list-style-type: none"> § Perlengkapan kerja diidentifikasi sesuai dengan kebutuhan data § Aplikasi beroperasi sesuai dengan standar aplikasi 	<ul style="list-style-type: none"> § Algoritma pemrograman 	<ul style="list-style-type: none"> § Menentu-kan data yang akan digunaka n sesuai dengan SOP 	<ul style="list-style-type: none"> § Mengurai-kan fungsi bermacam jenis data 	<ul style="list-style-type: none"> § Mengidenti-fikasi kebutuhan data

SUB KOMPETENSI	KRITERIA KINERJA	LINGKUP BELAJAR	MATERI POKOK PEMELAJARAN		
			SIKAP	PENGETAHUAN	KETRAMPILAN
3. Melakukan survey di lapangan	§ Prinsip kerja 4 struktur data pembentuk algoritma dan diagram alir (berurutan, pencabangan, pilihan, dan pengulangan / iterasi) § Prinsip kerja algoritma file berurutan § Prinsip kerja struktur data array dan matrik § Prinsip kerja metode pencarian (searching) § Prinsip kerja metode pengurutan (sorting) § Prinsip kerja penggabungan (merging) § Prinsip kerja metode pemrograman terstruktur	§ Metode pada algoritma pemrograman § Struktur data	§ Teliti dan cermat § Mengikuti prosedur survey sesuai SOP	§ Menyusun hasil survey di lapangan menjadi suatu acuan pembuatan program	§ Menyelesaikan survey di lapangan
4. Mengisi log sheet / report sheet	§ Log sheet/ report sheet diisi sesuai dengan SOP yang berlaku	§ Metode pada algoritma pemrograman § Struktur data	§ Disiplin dalam mengikuti SOP	§ Mengidentifikasi langkah-langkah pembuatan alur data dan diagram alir	§ Mempraktekkan pembuatan alur data dan diagram alir program

F. CEK KEMAMPUAN

Untuk mengetahui kemampuan awal yang telah Anda miliki, maka isilah cek lis (√) seperti pada tabel di bawah ini dengan sikap jujur dan dapat dipertanggungjawabkan.

Sub Kompetensi	Pernyataan	Saya dapat Melakukan Pekerjaan ini dengan Kompeten		Bila Jawaban "Ya" Kerjakan
		Ya	Tidak	
1. Memahami prosedur survey penentuan kebutuhan data	1. Menggunakan fungsi dan operator algoritma pemrograman dan atau diagram alir			Tes Formatif 1
	2. Menjelaskan prosedur operasi fungsi dan operator pada algoritma pemrograman dan atau diagram alir			Tes Formatif 2
2. Mempersiapkan pokok-pokok kebutuhan data	1. Menjelaskan fungsi berbagai macam jenis data.			Tes Formatif 3
	2. Mengidentifikasi kebutuhan data sesuai dengan yang diinginkan			Tes Formatif 4
3. Melakukan survey di lapangan	1. Menjelaskan berbagai metode pada algoritma pemrograman			Tes Formatif 5
	2. Menjelaskan dan menggunakan struktur data			Tes Formatif 6
3. Mengisi log sheet/report sheet	Pembuatan diagram alur data dan diagram alir program			Tes Formatif 7

Apabila anda menjawab TIDAK pada salah satu pernyataan di atas, maka pelajarilah modul ini.

BAB II PEMELAJARAN

A. RENCANA PEMELAJARAN SISWA

Kompetensi : Menyiapkan dan melakukan survey untuk
menentukan kebutuhan data

Jenis Kegiatan	Tanggal	Waktu	Tempat Belajar	Alasan Perubahan	Tanda Tangan Guru
Konsep Algoritma dan Diagram Alir					
Algoritma dan Diagram Alir untuk Proses Pengulangan (Loop)					
Operasi Pada Struktur Data Larik 1 Dimensi					

B. KEGIATAN BELAJAR

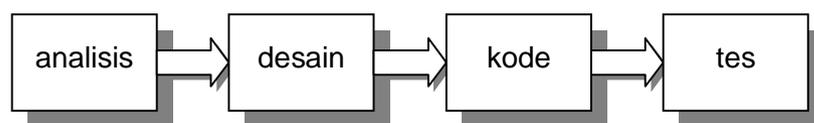
1. Kegiatan Belajar 1 : Konsep Algoritma dan Diagram Alir

a. Tujuan Kegiatan Pemelajaran

- 1) Peserta diklat mampu menerangkan konsep algoritma
- 2) Peserta diklat mampu menerangkan konsep diagram alir.
- 3) Peserta diklat mampu menggunakan konsep algoritma dan diagram alir untuk suatu kasus tertentu yang sederhana dalam kehidupan sehari-hari.

b. Uraian Materi 1

Dalam proses rekayasa perangkat lunak (RPL) dapat dipakai beberapa model pendekatan. Salah satu model rekayasa perangkat lunak yang paling tua (konvensional) dan sudah populer adalah model sekuensial linear yang sering disebut dengan "siklus kehidupan klasik" atau "model air terjun".



Gambar 1. Model Sekuensial Linier

Tahap paling awal pada proses RPL dengan pendekatan model sekuensial linier ini adalah tahapan analisis. Keberhasilan pada tahap ketiga kode (implementasi) sangat ditentukan oleh keberhasilan pada tahapan analisis dan desain.

Dalam tahap analisis yang dilakukan diantaranya adalah melakukan survey kebutuhan misalnya: mengidentifikasi dan menentukan kebutuhan data. Lalu pada tahap desain dilakukan pemodelan dari langkah-langkah aliran data dan atau aliran program, hingga

mencakup penentuan tipe data, struktur data, antarmuka input output dan penentuan bahasa pemrograman yang dipakai.

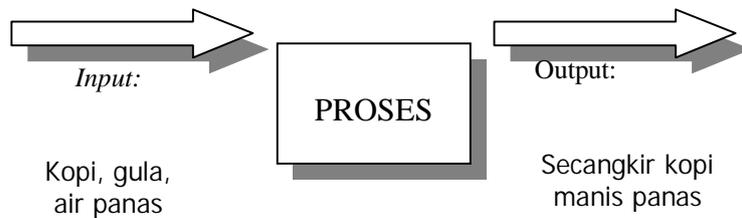
Untuk lingkup pekerjaan yang sederhana tahapan analisis dan desain ini bisa dilakukan dengan pendekatan algoritma pemrograman dan diagram alir data (program).

Konsep Algoritma

Algoritma dalam konteks umum adalah suatu urutan langkah pekerjaan yang ditulis dalam notasi deskriptif yang logis untuk pencapaian suatu solusi atau untuk menuju ke suatu tujuan tertentu. Untuk memudahkan pemahaman tentang algoritma ini dipakai suatu contoh kasus, misalnya diinginkan untuk membuat secangkir kopi manis, maka algoritmanya adalah:

- 1) Mulai
- 2) Siapkan kopi dan gula pasir
- 3) Siapkan satu cangkir bersih dan satu sendok makan
- 4) Masukkan sepucuk sendok makan kopi ke dalam cangkir
- 5) Masukkan satu setengah sendok makan gula pasir ke dalam cangkir
- 6) Tuangkan air panas ke dalam cangkir hingga hampir penuh
- 7) Aduk-aduk hingga semua gula larut dan tercampur
- 8) Selesai

Hal penting selanjutnya yang harus dipikirkan adalah menentukan kebutuhan data. Jika dipandang ketujuh langkah algoritma di atas adalah membangun suatu sistem sederhana, maka dapat diidentifikasi kebutuhan data untuk masukan ke sistem (input) adalah : kopi, gula, dan air panas, serta data untuk keluaran dari sistem tersebut adalah secangkir kopi manis panas.



Gambar 2. Gambaran Suatu Sistem Sederhana

Dari satu contoh sederhana di atas dapat dikembangkan sendiri algoritma-algoritma untuk contoh kasus yang lain.

Sementara itu yang disebut dengan program secara umum, adalah kumpulan instruksi atau perintah yang disusun sedemikian rupa sehingga mempunyai urutan nalar yang tepat untuk menyelesaikan suatu persoalan. Jadi algoritma pemrograman bisa dijelaskan sebagai urutan-instruksi atau perintah yang disusun sedemikian rupa sehingga mempunyai urutan nalar yang tepat dan harus dipatuhi untuk menyelesaikan suatu persoalan.

Contoh suatu algoritma pemrograman yang sederhana adalah algoritma untuk menentukan kuadrat dari suatu sembarang bilangan yang diberikan. Di sini kita harus menentukan tujuan akhir program nanti apa?

Langkah 1 : Mulai.

Langkah 2 : Masukkan atau berikan sembarang bilangan.
Simpan dalam variabel x .

Langkah 3 : Kalikan bilangan itu (x) dengan dirinya sendirinya, dan simpan hasilnya di variabel y atau $y=x*x$.

Langkah 4 : Tampilkan hasilnya (y).

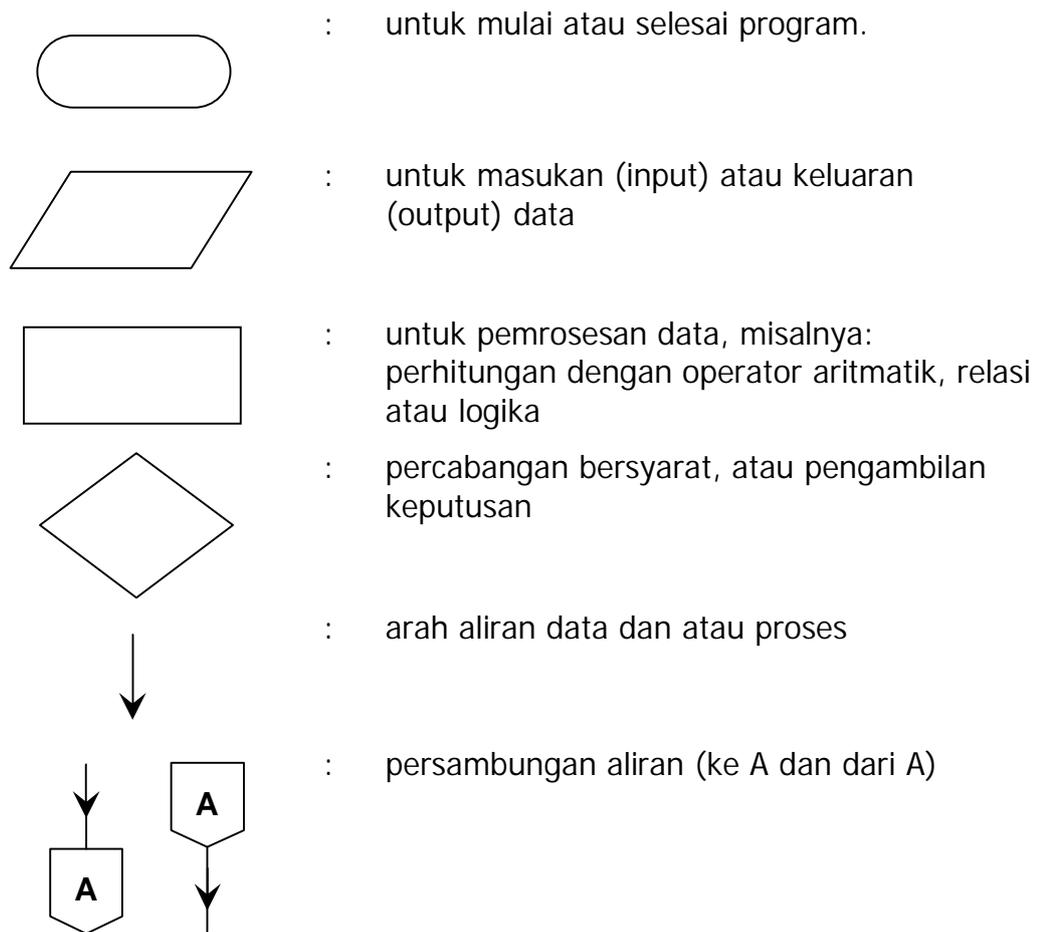
Langkah 5 : Selesai.

Sampai disini, langkah selanjutnya adalah mengidentifikasi dan menentukan kebutuhan dan atau jenis data yang dapat ditampung oleh variabel x (data masukan) dan variabel y (data keluaran), sehingga nantinya akan menentukan jenis variabel x dan y tersebut. Jelas bahwa jenis variabel y tergantung dari jenis variabel x karena y menampung hasil operasi perkalian dari variabel x , apabila x dapat menampung bilangan pecahan (real) maka y juga harus dapat menampung bilangan real, dengan kata lain jika x bertipe real (integer) maka y juga harus bertipe real. Tetapi jika x adalah bilangan bulat (integer) maka hasil kuadratnya pasti bilangan bulat juga, sehingga y bisa bertipe integer atau real.

Disamping untuk keperluan proses analisis dan desain, algoritma pemrograman juga berfungsi untuk memberikan kerangka berpikir yang lebih konkrit dan jelas kepada programmer sehingga dapat mempermudah pekerjaan pada proses coding.

Diagram Alir (Flow Chart)

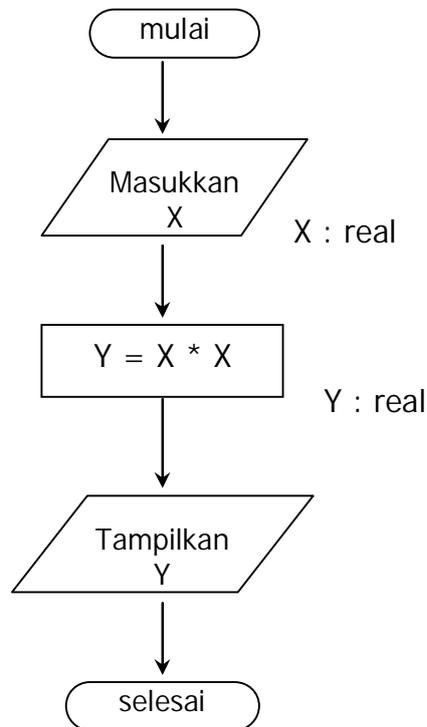
Sementara itu untuk konsep tentang diagram alir, hampir sama dengan konsep algoritma, hanya saja di sini penulisannya lebih disederhanakan yakni tidak dideskripsikan dengan kalimat-kalimat yang panjang tetapi dengan memakai simbol-simbol fungsi dan operator. Adapun simbol-simbol yang sering digunakan dalam penggambaran diagram alir adalah sebagai berikut.



Gambar 3. Simbol-simbol yang Digunakan dalam Diagram Alir

Ada beberapa operator yang digunakan yakni operator aritmatika dan operator relasi (himpunan). Diagram alir bisa merupakan representasi dari aliran data atau aliran program (urut-urutan instruksi). Tetapi biasanya suatu gambar diagram alir merupakan representasi keduanya, artinya suatu diagram alir bisa dipandang sebagai diagram aliran data sekaligus sebagai diagram aliran instruksi-instruksi program. Suatu diagram alir yang merepresentasikan aliran instruksi program dan data maka dapat dijadikan sebagai desain atau model dari perangkat lunak (program) anda.

Dalam contoh kasus algoritma pemrograman di atas dapat dibuat diagram alir-nya sebagai berikut.



Gambar 4. Diagram Alir Menentukan Kuadrat dari Sembarang Nilai X

Dari Gambar 4 di atas terlihat semakin jelas alur berpikir kita dan semakin mudah untuk menuangkannya dalam bentuk kode program. Kapan kita harus mulai proses pemasukkan data, lalu kapan harus melakukan proses pengolahan, dan kapan harus menampilkan hasil pengolahan tersebut, sudah terbayang dengan jelas.

Di sini juga dapat dilihat bahwa diagram alir di atas disamping merupakan diagram alir dari instruksi-instruksi program, juga sekaligus menunjukkan diagram alir data.

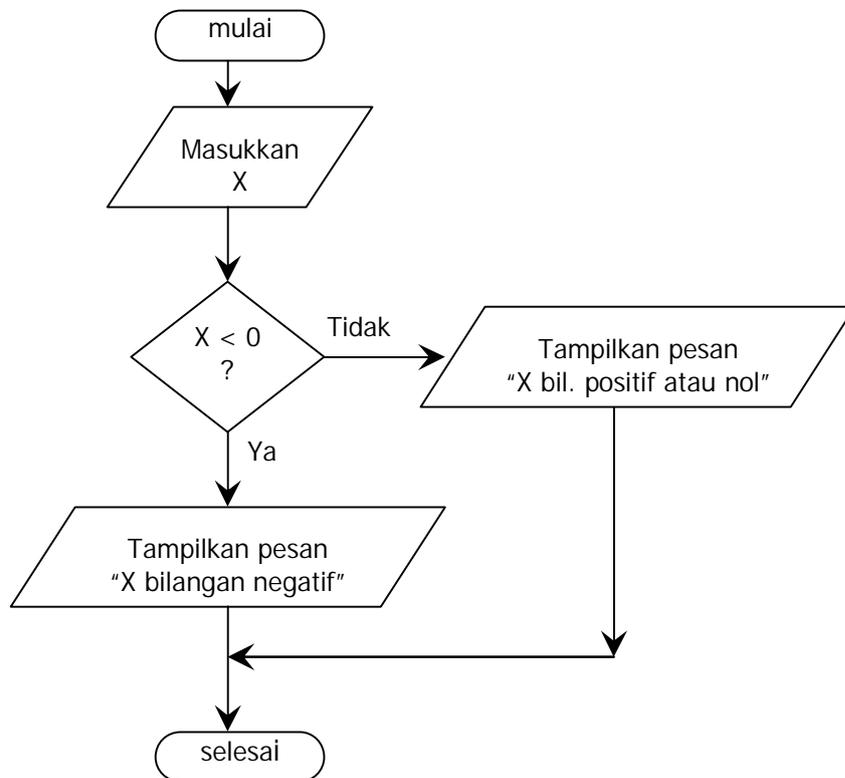
Proses Percabangan

Algoritma dan diagram alir yang telah dibahas di atas adalah menggunakan struktur data berurutan, artinya dari langkah pertama sampai dengan langkah terakhir tidak ada proses percabangan (pengambilan keputusan), pilihan ataupun pengulangan.

Sementara itu untuk proses percabangan, digunakan apabila di dalam prosesnya, urutan aliran program diharuskan untuk melompat atau pergi ke suatu langkah tertentu yang tidak berurutan, hal ini sering juga disebut dengan percabangan. Proses percabangan ini bisa dengan syarat (percabangan bersyarat) ataupun tidak (percabangan tidak bersyarat).

Untuk percabangan bersyarat, urutan-instruksi program akan melompat ke suatu langkah tertentu yang dituju, jika syarat yang diberikan telah terpenuhi. Biasanya dalam implementasinya nanti menggunakan instruksi IF (jika).... . Kemudian untuk percabangan yang tidak bersyarat urutan program akan langsung melompat ke suatu langkah tertentu yang dituju, apabila pengerjaan urutan-instruksi telah sampai pada instruksi pergi ke label... (goto label)

Contoh kasus penggunaan simbol percabangan bersyarat atau pengambilan keputusan, misalnya dalam kasus di atas, tujuan program tidak untuk menghitung $Y = X * X$ (atau X^2), tetapi untuk mengetahui apakah nilai X (nilai yang kita masukkan) itu bilangan positif atau negatif. Jika demikian berarti ada suatu instruksi pengetesan yang digunakan untuk persyaratan melompat atau pengambilan keputusan, jika $X \geq 0$ (nol atau positif) maka program akan menampilkan pesan "X positif atau nol" dan sebaliknya jika $X < 0$ maka akan ditampilkan pesan "X negatif". Gambar diagram alir selengkapnya ada di dalam Gambar 5.



Gambar 5. Contoh Diagram Alir Percabangan Bersyarat

c. Rangkuman 1

- 1) Algoritma pemrograman adalah urutan perintah yang disusun sedemikian rupa sehingga mempunyai urutan nalar yang tepat dan harus dipatuhi untuk menyelesaikan suatu persoalan.
- 2) Sedangkan diagram alir adalah penulisan algoritma pemrograman dengan menggunakan simbol-simbol, yang dapat memudahkan dalam memahami alur program secara menyeluruh.
- 3) Algoritma pemrograman dan diagram alir keduanya dapat digunakan untuk tahapan analisis dan desain dalam proses RPL yang sederhana.

d. Tugas 1

- 1) Pelajarilah uraian materi tentang algoritma pemrograman dan diagram alir ini!
- 2) Ada berapa macam jenis percabangan yang anda ketahui, jelaskan perbedaan masing-masing.
- 3) Buatlah algoritma pemrograman untuk diagram alir Gambar 4 di atas.
- 4) Identifikasi jenis data masukan dan data keluaran dari program yang dihasilkan dari diagram alir Gambar 5 di atas.

e. Tes Formatif 1

- 1) Gambarkan dan jelaskan fungsi simbol-simbol yang dipakai untuk membangun diagram alir!
- 2) Buatlah algoritma pemrograman untuk diagram alir dalam Gambar 5 di atas!
- 3) Diinginkan suatu program aplikasi yang bertujuan untuk mengetahui apakah suatu bilangan X yang dimasukkan oleh pengguna (user) dapat dibagi 3 atau tidak. Buatlah algoritma pemrograman dan diagram alirnya!

f. Kunci Jawaban Tes Formatif 1

- 1) Lihat Gambar 3 tentang simbol-simbol yang digunakan dalam diagram alir, halaman 12.

- 2) Langkah 1: Mulai.

Langkah 2: Masukkan atau berikan sembarang bilangan. Simpan dalam variabel X.

Langkah 3: Apakah $X < 0$?, jika ya : pergi ke langkah 4, jika tidak : pergi ke langkah 5.

Langkah 4: Tampilkan pesan "X bilangan negatif", pergi langkah 6.

Langkah 5: Tampilkan pesan "X bilangan positif atau nol".

Langkah 6: Selesai.

3) Langkah 1: Mulai.

Langkah 2: Masukkan atau berikan sembarang bilangan. Simpan dalam variabel X.

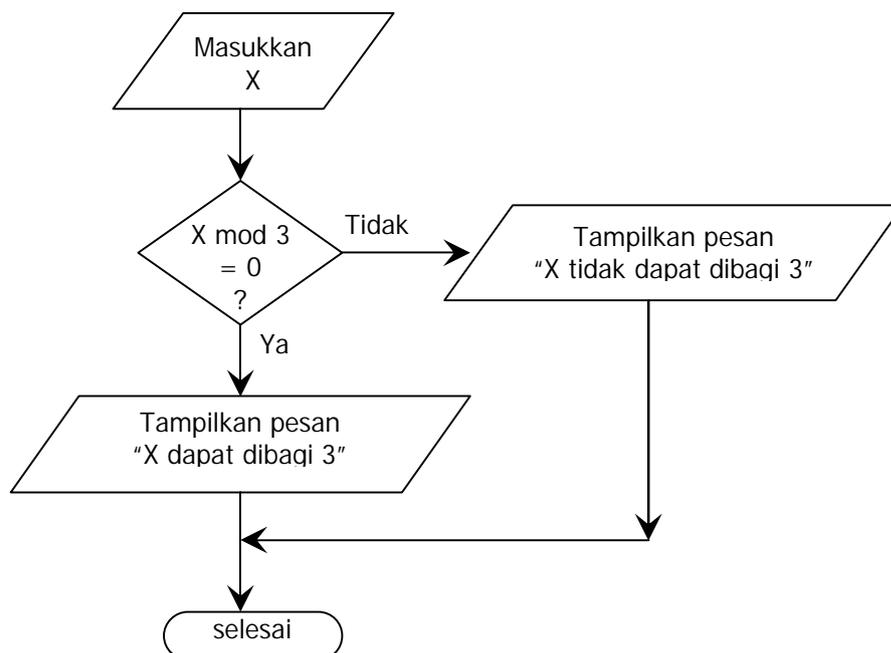
Langkah 3: Apakah $X \bmod 3 = 0$?, jika ya : pergi ke langkah 4, jika tidak : pergi ke langkah 5.

Langkah 4: Tampilkan pesan "X dapat dibagi 3", pergi langkah 6.

Langkah 5: Tampilkan pesan "X tidak dapat dibagi 3".

Langkah 6: Selesai.

Diagram alirnya:



Gambar 6. Contoh Diagram Alir Percabangan Bersyarat yang Lain

g. Lembar Kerja 1

Alat dan bahan :

Penggaris, alat tulis lengkap, dan kertas putih kosong 5 lembar.

Kesehatan dan Keselamatan Kerja

- 1) Gunakan peralatan sesuai dengan fungsinya.
- 2) Bekerjalah sesuai dengan cara kerja atau petunjuk yang telah ditentukan.

Langkah Kerja

- 1) Siapkan alat dan bahan yang digunakan.
- 2) Salinlah gambar 2 di atas kertas kosong.
- 3) Buatlah diagram alir di atas kertas kosong untuk contoh-contoh kasus di atas (Gambar 3, 4 dan 5)
- 4) Tuliskan langkah-langkah algoritma pemrogramannya di dekat (boleh di samping atau di bawah langsung) dari gambar simbol-simbol pada diagram alir tersebut.
- 5) Tuliskan nama dan nomor urut peserta diklat di pojok kanan bawah di setiap kertas kerja.
- 6) Periksa dan perbaiki gambar atau tulisan yang kurang sempurna atau salah.
- 7) Laporkan hasil pekerjaan anda pada guru pembimbing (pengajar).
- 8) Jika semua telah selesai rapikan alat dan bahan kemudian kembalikan ke tempat semula.

2. Kegiatan Belajar 2: Algoritma dan Diagram Alir untuk Proses Pengulangan (Looping)

a. Tujuan Kegiatan Pemelajaran

- 1) Peserta diklat mampu mengidentifikasi suatu permasalahan di lapangan yang mana permasalahan tersebut harus diselesaikan dengan cara perulangan (looping).
- 2) Peserta diklat mampu mengidentifikasi dan menentukan jenis dari semua kebutuhan data (baik data untuk masukan ataupun data keluaran).
- 3) Peserta diklat dapat membuat algoritma dan diagram alir untuk memecahkan masalah tersebut.

b. Uraian Materi 2

Dalam Kegiatan Belajar 1 telah dipelajari tentang konsep algoritma dan diagram alir pemrograman. Dalam Kegiatan Belajar 2 ini akan dipelajari lebih lanjut tentang penggunaan algoritma dan diagram alir pemrograman untuk mendesain program yang menggunakan proses pengulangan atau looping.

Loop adalah istilah tidak resmi untuk menyatakan segala jenis struktur kendali yang mengalami pengulangan (iterasi); yaitu, setiap struktur yang menyebabkan program berulang kali melaksanakan suatu blok kode. Memakai loop adalah salah satu aspek pemrograman yang paling kompleks dalam bahasa dengan aturan tertentu. Mengetahui cara dan kapan setiap loop itu dipakai sangat menentukan pembuatan software berkualitas tinggi.

Dalam sebagian besar bahasa pemrograman, ada beberapa jenis loop yakni; loop yang sudah ditentukan jumlah pelaksanaannya, misalnya satu kali untuk setiap pegawai. Loop yang dievaluasi secara kontinu. Untuk loop jenis ini tidak diketahui berapa kali pelaksanaannya, dan

setiap kali diulangi selalu diperiksa apakah itu adalah loop terakhir. Ada juga jenis loop yang tidak pernah berakhir.

Macam loop juga dibedakan oleh lokasi di mana pengujian kesempurnaan itu dilaksanakan. Pengujian dapat dilaksanakan di awal, tengah atau akhir loop. Karakteristik ini memberitahukan bahwa apakah sedikitnya loop itu dieksekusi satu kali. Jika pengujian dilakukan di awal, maka bagian utama loop tidak perlu dieksekusi. Jika loop diuji di akhir, bagian badan loop harus dieksekusi sedikitnya satu kali. Jika loop diuji ditengah, maka bagian loop yang mendahului bagian yang diuji, dieksekusi sedikitnya satu kali. Tapi bagian loop sesudahnya, yaitu yang menyusul bagian yang sudah diuji tidak perlu dieksekusi sama sekali.

Untuk mempermudah penjelasan dan pembelajaran tentang proses loop ini akan digunakan studi kasus, sebagai berikut. Diinginkan untuk membuat program yang dapat menampilkan semua bilangan genap antara 0 sampai dengan 100. Untuk menyelesaikan permasalahan itu maka harus dilakukan identifikasi masalah terlebih dahulu. Bilangan genap adalah bilangan yang habis dibagi dua, sehingga untuk bilangan genap antara 1 –100 adalah 2,4,6,8, .. dst. Hal ini akan lebih efektif jika dipecahkan dengan memakai loop. Adapun algoritmanya adalah sebagai berikut:

Langkah 1 : Mulai

Langkah 2 : Tentukan $i=1$

Langkah 3 : Apakah $i \bmod 2 = 0$, jika ya (i bilangan genap), maka pergi ke langkah 4, tetapi jika tidak, maka pergi ke langkah 5

Langkah 4 : Tampilkan i

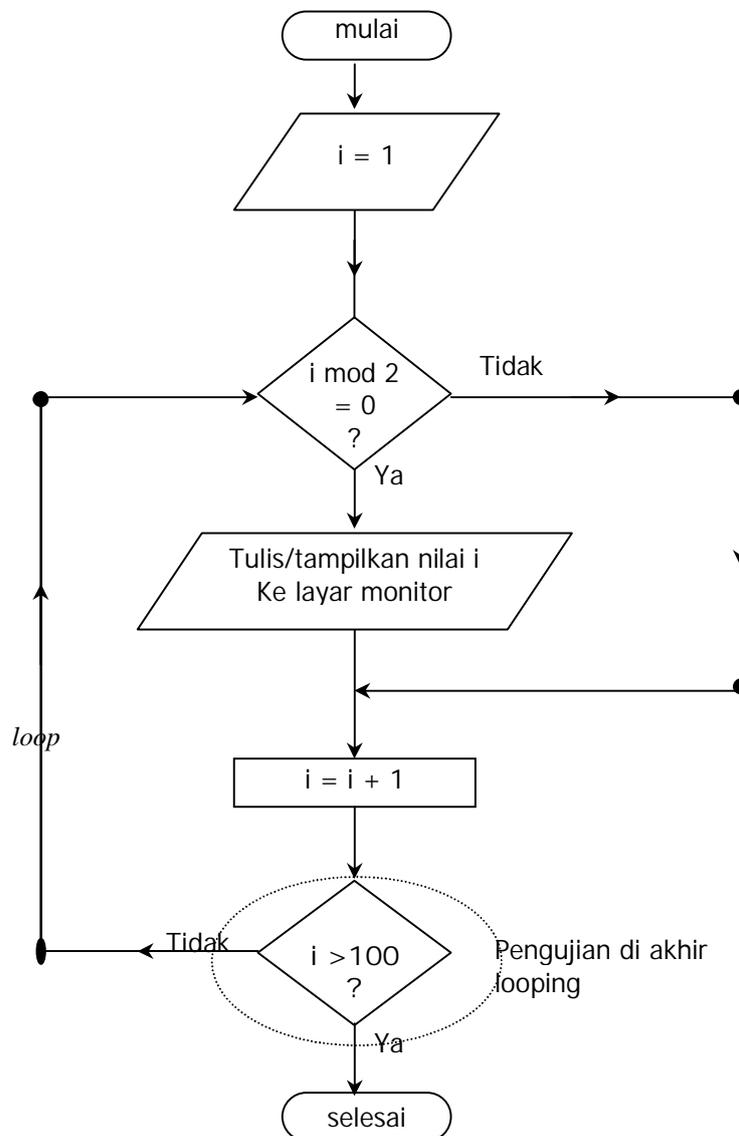
Langkah 5 : Tentukan $i = i + 1$

Langkah 6 : Apakah $i > 100$ jika ya, pergi ke langkah 7,
jika tidak maka kembali ke langkah 3

Langkah 7 : Selesai

Dalam langkah 3 algoritma di atas terdapat operator *Mod*. *Mod* adalah suatu operator aritmatika yang digunakan untuk menghitung sisa hasil bagi antara dua buah bilangan bulat atau integer. Hasil dari operasi mod ini adalah suatu bilangan integer juga.

Kemudian untuk diagram alir programnya adalah sebagai berikut:



Gambar 7. Diagram alir untuk proses looping (pengujian di akhir)

Di sini terlihat bahwa loop yang dipakai adalah jenis loop dengan pengujian di akhir, artinya bagian badan loop harus dieksekusi sedikitnya satu kali.

c. Rangkuman 2

- 1) Loop yang biasa digunakan berdasarkan letak pengujiannya ada tiga macam: loop dengan pengujian di awal, di akhir dan di tengah.
- 2) Sangat penting untuk diketahui cara dan kapan suatu jenis loop itu dipakai, karena hal ini sangat menentukan pembuatan software yang berkualitas tinggi.

d. Tugas 2

- 1) Buatlah algoritma dan diagram alir untuk kasus di atas, tetapi dengan menggunakan loop pengujian di awal.
- 2) Buatlah algoritma dan diagram alir untuk kasus diatas, tetapi dengan menggunakan loop pengujian di tengah.
- 3) Berikan komentar dan kesimpulan dari hasil pekerjaan item 1 dan 2 di atas.

e. Tes Formatif 2

- 1) Sebutkan dan jelaskan macam-macam loop berdasarkan letak pengujiannya!
- 2) Buatlah algoritma dan diagram alir program untuk menampilkan seluruh bilangan yang habis dibagi 3 dan juga habis dibagi 5, antara 1-100!
- 3) Buatlah algoritma dan diagram alir program untuk menghitung jumlah bilangan ganjil antara 1-100!
- 4) Identifikasi semua kebutuhan data (data masukan dan data keluaran) untuk kasus nomor 2 dan 3 di atas!

f. Kunci Jawaban Tes Formatif 2

- 1) Sudah dijelaskan dalam seksi Uraian Materi 2 di atas.
- 2) Algoritma untuk menampilkan seluruh bilangan antara 1-100 yang habis dibagi 3 dan juga habis dibagi 5.

Langkah 1 : Mulai

Langkah 2 : Tentukan $i=1$

Langkah 3 : Apakah $i \bmod 3 = 0$, jika ya (i habis dibagi 3), maka teruskan ke langkah 4 berikut, tetapi jika tidak, maka lompat ke langkah 6

Langkah 4 : Apakah $i \bmod 5=0$, jika ya (i juga habis dibagi 5), maka terus ke langkah 5, tetapi jika tidak, maka lompat ke langkah 6

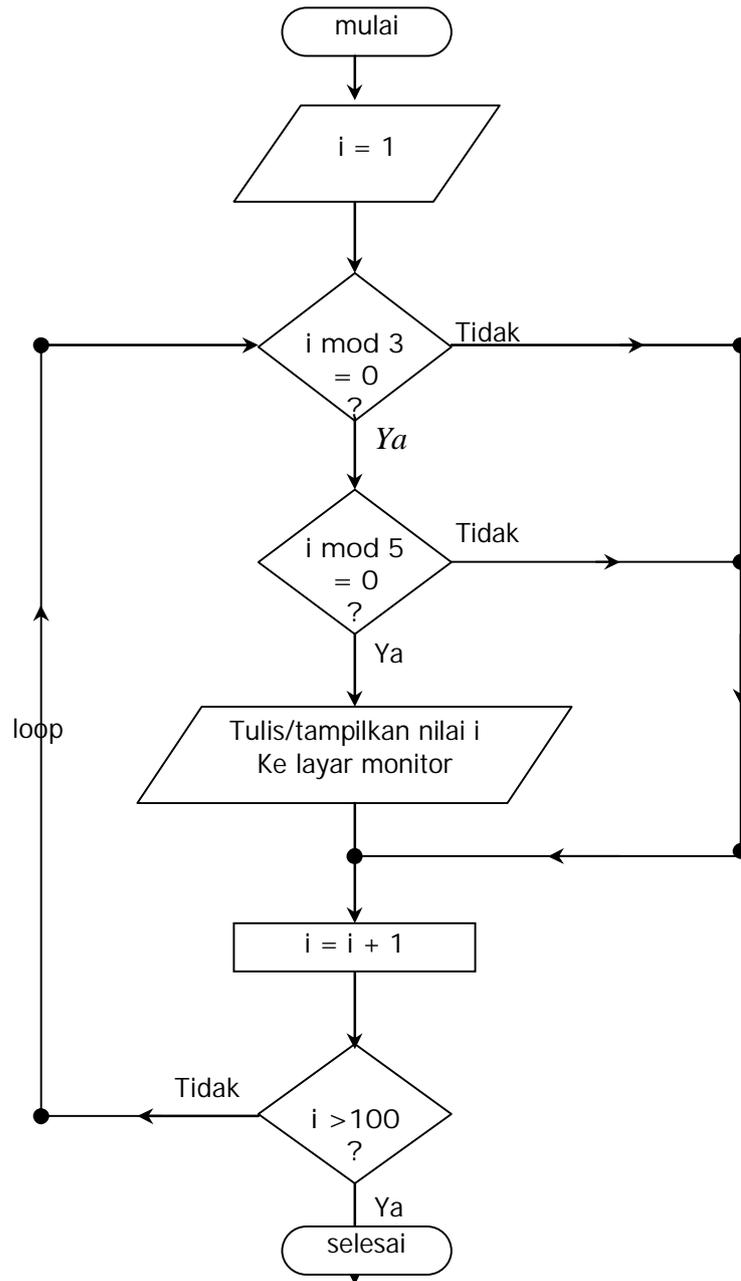
Langkah 5 : Tampilkan i

Langkah 6 : Tentukan $i = i + 1$

Langkah 7 : Apakah $i > 100$ jika ya, pergi ke langkah 8, jika tidak, maka kembali ke langkah 3

Langkah 8 : Selesai

Diagram alirnya adalah seperti gambar berikut ini.



Gambar 8. Diagram alir jawaban tes formatif-2 nomor 2

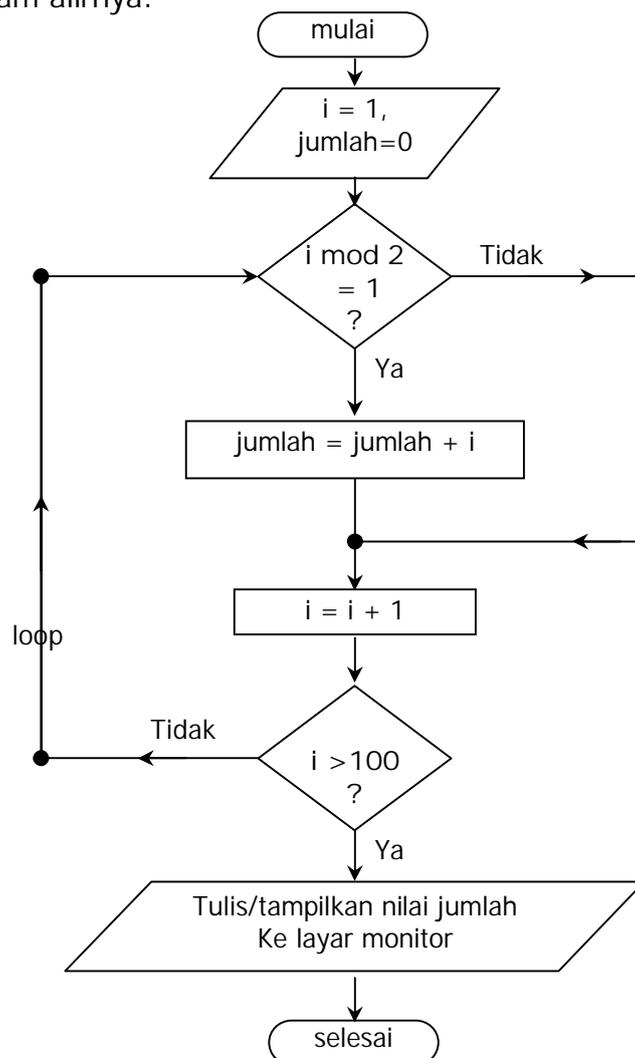
3) Algoritma menghitung jumlah bilangan ganjil antara 1-100.

Langkah 1 : Mulai

Langkah 2 : Tentukan $i=1$, $jumlah=0$;

- Langkah 3 : Apakah $i \bmod 2 = 1$, jika ya (i bilangan ganjil), maka terus ke langkah 4 berikut, tetapi jika tidak, maka lompat ke langkah 5
- Langkah 4 : Tentukan $\text{jumlah} := \text{jumlah} + i$;
- Langkah 5 : Tentukan $i = i + 1$
- Langkah 6 : Apakah $i > 100$ jika ya, langsung ke langkah 7, jika tidak, maka kembali ke langkah 3 (loop).
- Langkah 7 : Tampilkan pesan "Jumlah bilangan ganjil antara 1-100 = ", Tampilkan isi peubah jumlah.
- Langkah 8 : Selesai

Diagram alirnya:



Gambar 9. Diagram alir untuk soal tes formatif 2 nomor 3

4) Untuk soal nomor 2 dan 3, kebutuhan datanya adalah: data bilangan bulat antara 1-100. Bilangan ini tidak disimpan dalam suatu variabel namun, dibangkitkan dan diproses langsung ketika program berjalan. Data output, untuk yang nomor 2: bilangan antara 1-100 yang habis dibagi 3 dan habis pula dibagi 5 yakni: 15, 30, 45, 60, 75, dan 90.

Sementara itu untuk nomor 3 data outputnya adalah sebuah bilangan bulat (integer) yang merupakan hasil penjumlahan seluruh bilangan ganjil antara 1-100 = $1 + 3 + 5 + 7 + 8 + \dots + 95 + 97 + 99$.

g. Lembar Kerja 2

Alat dan bahan :

Penggaris, alat tulis lengkap, dan kertas putih kosong 5 lembar.

Kesehatan dan Keselamatan Kerja

- 1) Gunakan peralatan sesuai dengan fungsinya.
- 2) Bekerjalah sesuai dengan cara kerja atau petunjuk yang telah ditentukan.

Langkah Kerja

- 1) Siapkan alat dan bahan yang digunakan.
- 2) Kerjakan tugas 2 dan tes formatif 2 di atas dengan kertas kosong yang telah disediakan.
- 3) Pada kasus tes formatif-2 nomor 3 di atas menggunakan loop pengujian akhir, bagaimanakah gambar diagram alir dan algoritmanya apabila menggunakan jenis loop pengujian awal dan tengah.
- 4) Tuliskan nama dan nomor urut peserta diklat di pojok kanan bawah di setiap kertas kerja.

- 5) Periksa dan perbaiki gambar atau tulisan yang kurang sempurna atau salah.
- 6) Laporkan hasil pekerjaan anda pada guru pembimbing (pengajar).
- 7) Jika semua telah selesai rapikan alat dan bahan kemudian kembalikan ke tempat semula.

3. Kegiatan Belajar 3 : Operasi Pada Struktur Data Larik 1 Dimensi

a. Tujuan Kegiatan Pemelajaran

- 1) Peserta diklat dapat mengidentifikasi berbagai macam permasalahan yang ada di lapangan yang berkaitan dengan operasi larik.
- 2) Peserta diklat mampu mengidentifikasi dan membuat algoritma dan diagram alir suatu permasalahan untuk menentukan pencarian (searching) dan pengurutan (sorting).

b. Uraian Materi 3

Selama ini kita hanya mempelajari struktur data tunggal artinya data masukan yang akan diproses oleh program dan data keluaran sebagai hasil proses program bersifat tunggal (tidak berkelompok). Padahal permasalahan yang dijumpai di lapangan banyak jenis data yang ternyata teridentifikasi tepat apabila disimpan dengan peubah (variabel) dengan struktur data larik (array). Larik adalah kumpulan data yang sejenis dengan masing-masing elemen anggotanya memiliki sebuah nomor indek elemen yang unik. Contoh data yang tepat disimpan dengan struktur data larik ini adalah daftar nilai siswa. Apabila tidak disimpan dalam struktur data larik maka bila ada 30 siswa, diperlukan deklarasi 30 peubah unik untuk menyimpan masing-masing nilai siswa tersebut. Tetapi apabila dengan struktur data larik cukup dideklarasikan satu peubah jenis larik dengan indek 1 sampai dengan 30, sebagai berikut :

```
Var nilai : array[1..30] of real;
```

nilai adalah nama peubah bertipe larik.

Untuk memberi nilai variabel nilai[1] sampai dengan nilai[30] cukup dengan cara sebagai berikut:

```

nilai[1] :=67.5;      (nilai untuk siswa ke-1)
nilai[2] :=80;      (nilai untuk siswa ke-2)
.
.
.
nilai[30]:=75,5;    (nilai untuk siswa ke-30)

```

Sampai di sini struktur data larik yang telah kita pelajari di atas adalah larik satu dimensi. Struktur data larik yang lebih kompleks lagi adalah struktur data larik 2 dimensi. Larik 2 dimensi biasanya dipakai untuk menyimpan data dari suatu matrik atau tabel, yang mana diperlukan informasi nomor baris dan kolom untuk menunjuk tiap elemennya. Sehingga dalam struktur data larik 2 dimensi tiap elemen mempunyai 2 buah indek, satu untuk indek nomor baris dan satu lagi untuk indek nomor kolom. Misalnya untuk menyimpan data dari semua elemen matrik berukuran 3x4, maka deklarasi peubahnya adalah sebagai berikut:

```

Var matrik : array[1..3][1..4] of real;

```

matrik adalah nama peubah bertipe larik 2 dimensi. Untuk pengisian nilai ke variabel larik 2 dimensi ini sama dengan larik 1 dimensi, misalnya : matrik[1][3]:=5,75;

Pencarian Data (Searching)

Mencari suatu data atau searching pada sekelompok data yang sejenis adalah suatu proses yang sering dijumpai dilapangan. Misalnya dalam contoh kasus larik 1 dimensi yang digunakan untuk menyimpan nilai dari 30 siswa di atas, diinginkan mencari siswa yang nilainya sama dengan 80, atau mencari semua siswa yang nilainya > 60. Karena proses pencarian ini biasa terjadi pada sekelompok data yang sejenis, maka biasanya proses ini berkaitan langsung dengan struktur data larik. Prinsip kerja dari metoda searching ini adalah sebagai berikut: membandingkan mulai dari data (indek) pertama

hingga data (indek) terakhir pada peubah larik tersebut dengan data yang dicari, jika ada yang sama berarti proses pencarian sukses atau ketemu, tetapi jika sampai data yang terakhir tidak ada yang sama berarti proses pencarian gagal atau tidak ditemukan data yang dimaksud.

Dalam contoh larik nilai dari 30 siswa di atas, algoritma untuk proses searching ini adalah sebagai berikut.

Langkah 1: Mulai.

Langkah 2: Masukkan data yang akan dicari, simpan dalam variabel X.

Langkah 3: Ketemu:=0; i:=1. (inisialisasi variabel ketemu dan penghitung jumlah loop)

Langkah 4: Apakah nilai[i]=X, jika ya: set variabel Ketemu:=1 lalu pergi ke langkah 7; jika tidak, pergi ke langkah 5 berikut.

Langkah 5: Apakah i=30, jika ya pergi ke langkah 7, jika tidak pergi ke langkah 6 berikut.

Langkah 6: i=i+1; kembali ke langkah 4.

Langkah 7: Apakah ketemu=1? Jika ya langsung ke langkah 8 berikut ini, jika tidak, pergi ke langkah 9.

Langkah 8: Tampilkan pesan "Proses pencarian berhasil. Data X ditemukan, yakni pada elemen nilai ke i". Lalu pergi ke langkah 10.

Langkah 9: Tampilkan pesan "Proses pencarian gagal (data tidak ditemukan)"

Langkah 10: Selesai.

Peubah ketemu dalam algoritma di atas dipakai untuk pengecekan selama proses searching berlangsung. Jika status ketemu=0 berarti data tidak/belum ketemu tetapi jika =1 berarti data ditemukan. Pada

awalnya diasumsikan data tidak ketemu sehingga ketemu diset nol, jika ditemukan data yang dicari maka variabel ketemu ini langsung diubah statusnya menjadi 1, baru pergi ke langkah selanjutnya. Apabila sampai akhir proses perbandingan status ketemu masih sama dengan nol, berarti tidak ditemukan data yang dicari dalam variabel larik tersebut.

Pengurutan Data (Sorting)

Proses lain yang tak kalah pentingnya dan sering dijumpai dilapangan adalah proses mengurutkan data dalam suatu variabel larik (dalam hal ini larik 1 dimensi).

Pengurutan data ada dua macam yakni urut naik (ascending) dan urut turun (decending). Dalam urut naik, data dalam larik akan ditata ulang dengan elemen yang nilainya terkecil akan ditempatkan pada posisi nomor indek yang pertama dan elemen dengan nilai terbesar akan ditempatkan pada posisi nomor indek yang terakhir.

Ada beberapa metode pengurutan yang dikenal saat ini, tetapi yang sederhana dan relatif mudah diimplementasikan adalah metoda Buble Sort termodifikasi. Menurut Insap Santosa (1992), Metoda Buble Sort ini metoda yang paling tidak efisien karena banyak sekali langkah-langkah perbandingan data dan penukaran. Namun dengan memperhatikan algoritma metode pengurutan Seleksi, maka metoda Buble Sort dapat dimodifikasi sehingga langkah-langkahnya menjadi lebih efisien.

Prinsip dari metoda ini adalah sebagai berikut (pada urut turun/decending): dimulai dari data pertama (elemen larik pada nomor indek yang pertama) dari larik 1 dimensi di atas sebagai patokan, lalu dibandingkan dengan data kedua (elemen larik pada nomor indek yang kedua), apabila data kedua $>$ data pertama, maka

kedua data tersebut dipertukarkan sehingga data kedua menjadi data pertama dan data pertama menjadi data kedua, selanjutnya data pertama dibandingkan lagi dengan data ketiga (elemen larik pada nomor indek yang ketiga), apabila data ketiga $>$ data pertama maka keduanya ditukarkan, demikian seterusnya hingga data terakhir (elemen larik pada nomor indek yang terakhir) dibandingkan dengan data pertama. Perbandingan dilanjutkan dengan data kedua sebagai patokannya, dan dibandingkan dengan data ketiga dan seterusnya hingga data terakhir. Demikian algoritma berulang terus hingga data $n-1$ (dimana n adalah jumlah elemen larik seluruhnya, yang mana dalam contoh ini $n=30$) dipakai sebagai patokannya.

Dalam contoh larik nilai dari 30 siswa di atas, algoritma untuk proses pengurutan (ascending) adalah sebagai berikut.

Langkah 1: Mulai

Langkah 2: Tentukan $i := 1$;

Langkah 3: Tentukan $j := i + 1$;

Langkah 4: Apakah $nilai[i] > nilai[j]$? Jika tidak, pergi ke langkah 6, Jika ya lanjutkan ke langkah 5 berikut ini.

Langkah 5: Pertukarkan nilai dari $nilai[i]$ dengan $nilai[j]$.

Pseudocode-nya adalah sebagai berikut :

```
titip := nilai[i];
nilai[i] := nilai[j];
nilai[j] := titip;
```

Langkah 6: Tentukan $j := j + 1$; Apakah $j > n$ (dalam kasus ini $j > 30$)? Jika tidak, pergi ke langkah 4, Jika ya langsung ke langkah 7 berikut.

Langkah 7: Tentukan $i := i + 1$; Apakah $i > n - 1$ (dalam kasus ini $j > 29$)? Jika tidak, pergi ke langkah 3, Jika ya pergi ke langkah 8 berikut.

Langkah 8: Selesai.

c. Rangkuman 3

- 1) Larik (array) adalah struktur data yang terdiri atas sekumpulan data yang sejenis. Larik ada yang 1 dimensi dan 2 dimensi.
- 2) Beberapa operasi penting pada struktur data larik adalah pencarian data dalam larik, serta pengurutan data dalam larik.
- 3) Pengurutan (sorting) ada dua macam:urut naik (ascending) dan urut turun (decending).
- 4) Salah satu metoda sorting yang mudah untuk diimplementasikan adalah metoda buble sort termodifikasi.

d. Tugas 3

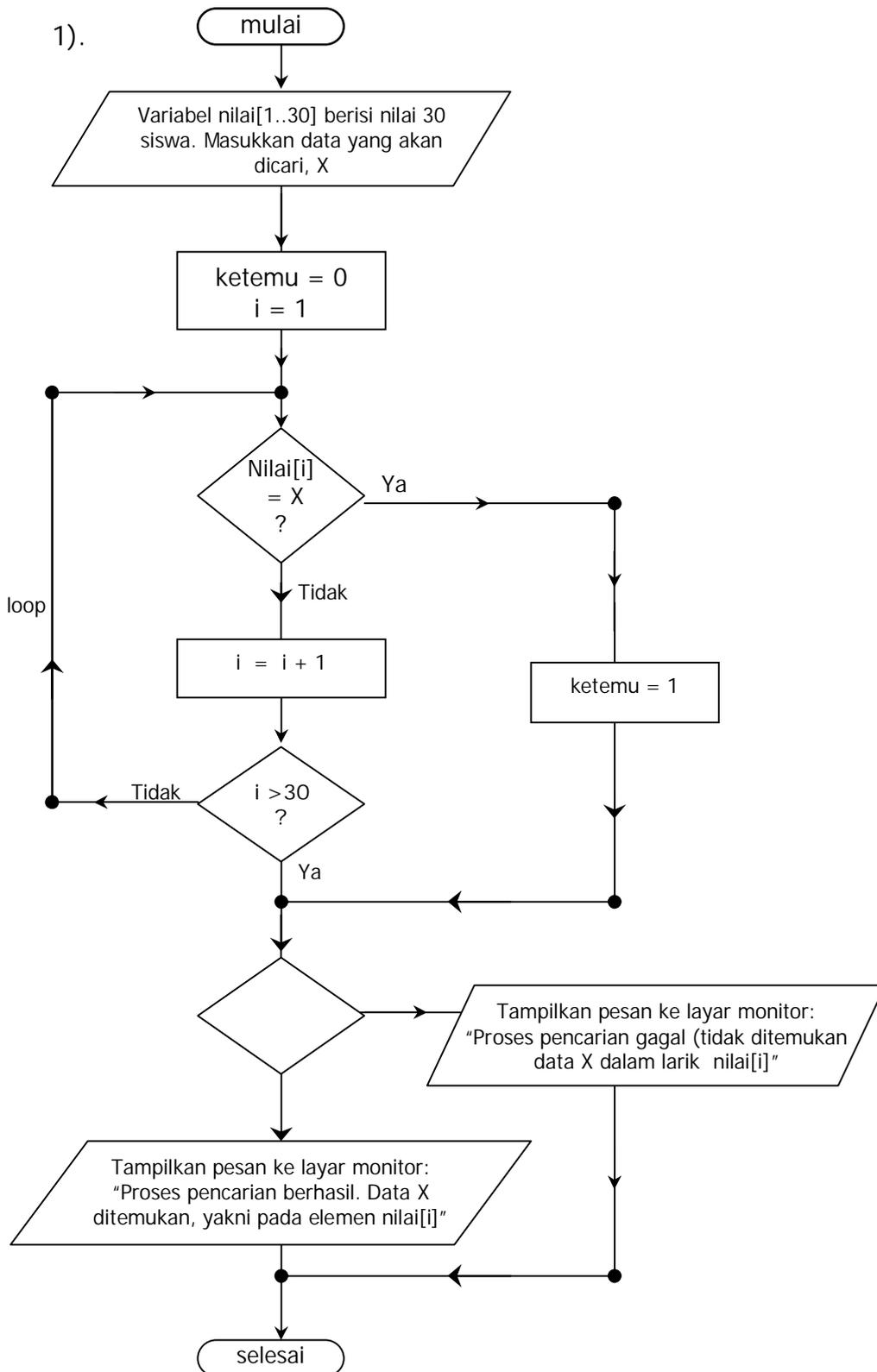
- 1) Bacalah dan pelajari sekali lagi keseluruhan uraian materi di atas, terutama untuk prinsip kerja pengurutan data dengan metode buble sort termodifikasi di atas.
- 2) Carilah sumber bahan bacaan atau buku referensi yang menjelaskan tentang metode pengurutan buble sort (yang belum termodifikasi) dan metode seleksi, seperti telah disinggung dalam uraian materi 3 di atas.

e. Tes Formatif 3

- 1) Buatlah diagram alir untuk algoritma pencarian data di atas!
- 2) Buatlah diagram alir untuk algoritma pengurutan data di atas!
- 3) Bagaimana caranya mengubah algoritma pengurutan ascending (urut naik) di atas menjadi algoritma pengurutan decending (urut turun)?
- 4) Buatlah algoritma pemrograman untuk menghitung nilai rata-rata dari seluruh 30 siswa dalam contoh di atas!

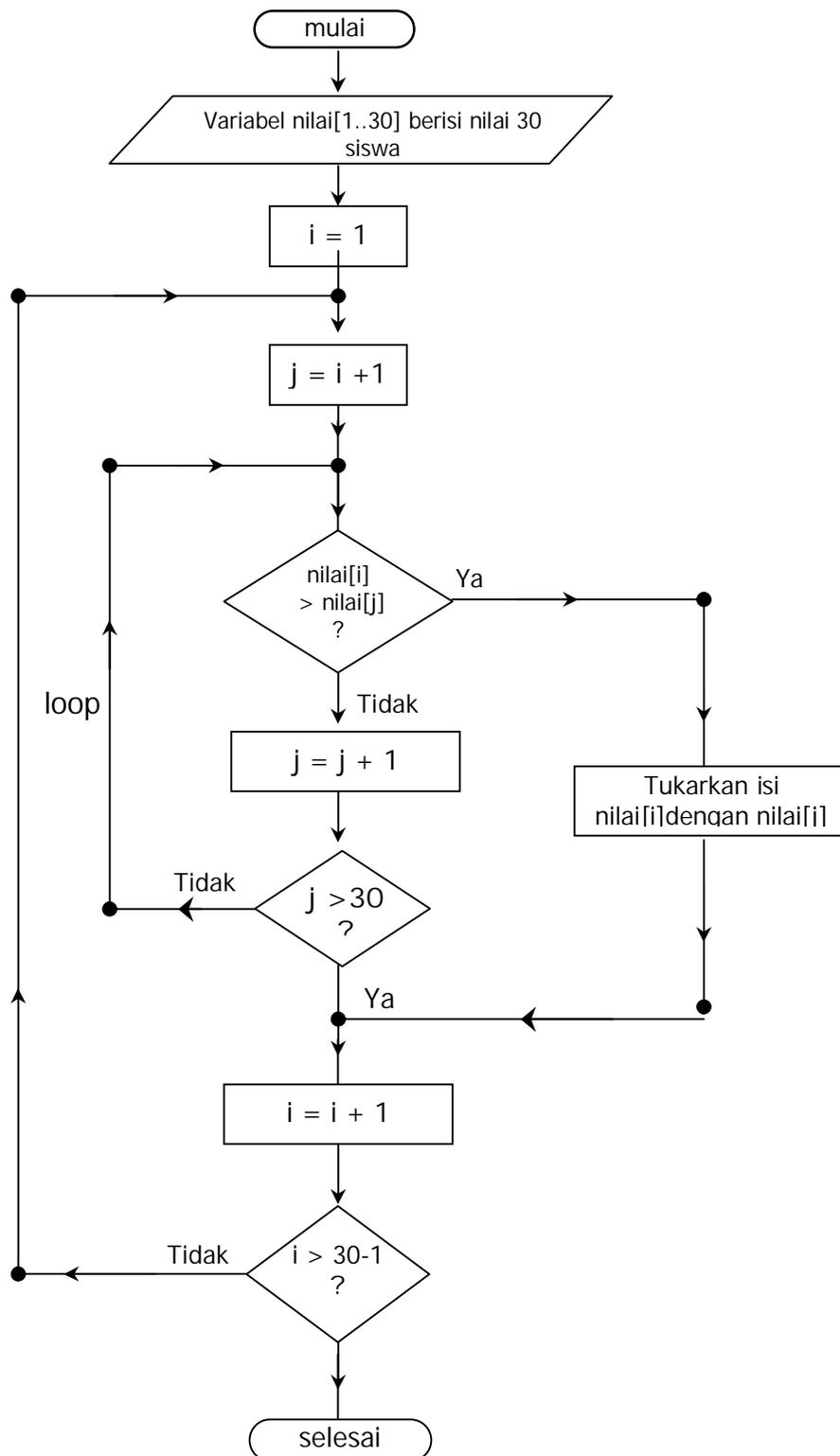
f. Kunci Jawaban Tes Formatif 3

1).



Gambar 10. Diagram alir untuk soal tes formatif 3 nomor 1

2).



Gambar 11. Diagram alir untuk soal tes formatif 3 nomor 2

3). Pada langkah 4 gantilah tanda lebih besar(">") dengan lebih kecil ("<") sehingga menjadi: Apakah `nilai[i] < nilai[j]`? Jika tidak, pergi ke langkah 6, Jika ya lanjutkan ke langkah 5 berikut ini.

4). Algoritma pemrograman untuk menghitung nilai rata-rata:

Langkah 1: Mulai

Langkah 2: Tentukan `titip:=0; i:=1;`

Langkah 3: `titip := titip + i;`

Langkah 4: Tentukan `i := i + 1;`

Langkah 5: Apakah `i>30`? Jika tidak, pergi ke langkah 3, Jika ya lanjutkan ke langkah 6.

Langkah 6: Selesai.

g. Lembar Kerja 3

Alat dan bahan :

Penggaris, alat tulis lengkap, dan kertas putih kosong 5 lembar.

Kesehatan dan Keselamatan Kerja

- 1) Gunakan peralatan sesuai dengan fungsinya.
- 2) Bekerjalah sesuai dengan cara kerja atau petunjuk yang telah ditentukan.

Langkah Kerja

- 1) Siapkan alat dan bahan yang digunakan.
- 2) Kerjakan tugas 3 dan tes formatif 3 di atas dengan kertas kosong yang telah disediakan.
- 3) Kelas A dan B masing-masing berisi 30 siswa. Nilai matematika kelas A disimpan pada variabel `matA[1..30]` dan kelas B disimpan pada variabel `matB[1..30]`. Apabila diinginkan untuk

menjumlahkan nilai dari 2 variabel tersebut buatlah algoritma dan diagram alir pemrogramannya!

- 4) Tuliskan nama dan nomor urut peserta diklat di pojok kanan bawah di setiap kertas kerja.
- 5) Periksa dan perbaiki gambar atau tulisan yang kurang sempurna atau salah.
- 6) Laporkan hasil pekerjaan anda pada guru pembimbing (pengajar).
- 7) Jika semua telah selesai rapikan alat dan bahan kemudian kembalikan ke tempat semula.

BAB III

EVALUASI

A. PERTANYAAN

- 1) Jelaskan apa yang dimaksud dengan algoritma dan diagram alir itu?
- 2) Apakah yang disebut dengan program itu?
- 3) Sebutkan dan jelaskan tahap-tahap yang harus dilalui dalam rekayasa perangkat lunak dengan pendekatan metode konvensional (model air terjun)!
- 4) Buatlah algoritma dan diagram alir pemrograman untuk menghitung dan atau menemukan bilangan prima antara 2 sampai 50.
- 5) Buatlah algoritma dan diagram alir pemrograman untuk proses mencari data maksimum dan minimum dari suatu larik 1 dimensi. Gunakan larik data seperti kegiatan belajar di atas.

B. KUNCI JAWABAN EVALUASI

1. Lihat pada seksi Uraian Materi 1 hal. 7 –10.
2. Urut-urutan instruksi yang dimengerti oleh komputer untuk tujuan tertentu.
3. Lihat pada seksi Uraian Materi 1 hal. 7 –10.
4. Algoritma menghitung/menemukan bilangan prima antara 1 –50

Langkah 1: Mulai

Langkah 2: Tentukan $i:=2$;

Langkah 3: Tentukan (diasumsikan) bilangan i adalah prima, sehingga set awal variabel `prima:=True`

Langkah 4: Tentukan $j:=i-1$;

Langkah 5: Cek apakah i habis dibagi j , jika ya pergi ke langkah 6, jika tidak pergi ke langkah 7

- Langkah 6: berarti bukan bilangan prima, set prima:=False;
- Langkah 7: $j:=j-1$; Apakah $j=2$? jika tidak kembali ke langkah 5, jika ya terus ke langkah 8 berikut.
- Langkah 8: Cek apakah prima=True? Jika ya, tampilkan i, jika tidak terus ke langkah 9.
- Langkah 9: $i := i + 1$, Apakah $i = 50$? jika tidak kembali ke langkah 3, jika ya pergi ke langkah 9.
- Langkah 10: Selesai.

Untuk memperjelas keterangan di atas, berikut diberikan pseudocode untuk algoritma menemukan bilangan prima 2-50 diatas.

```

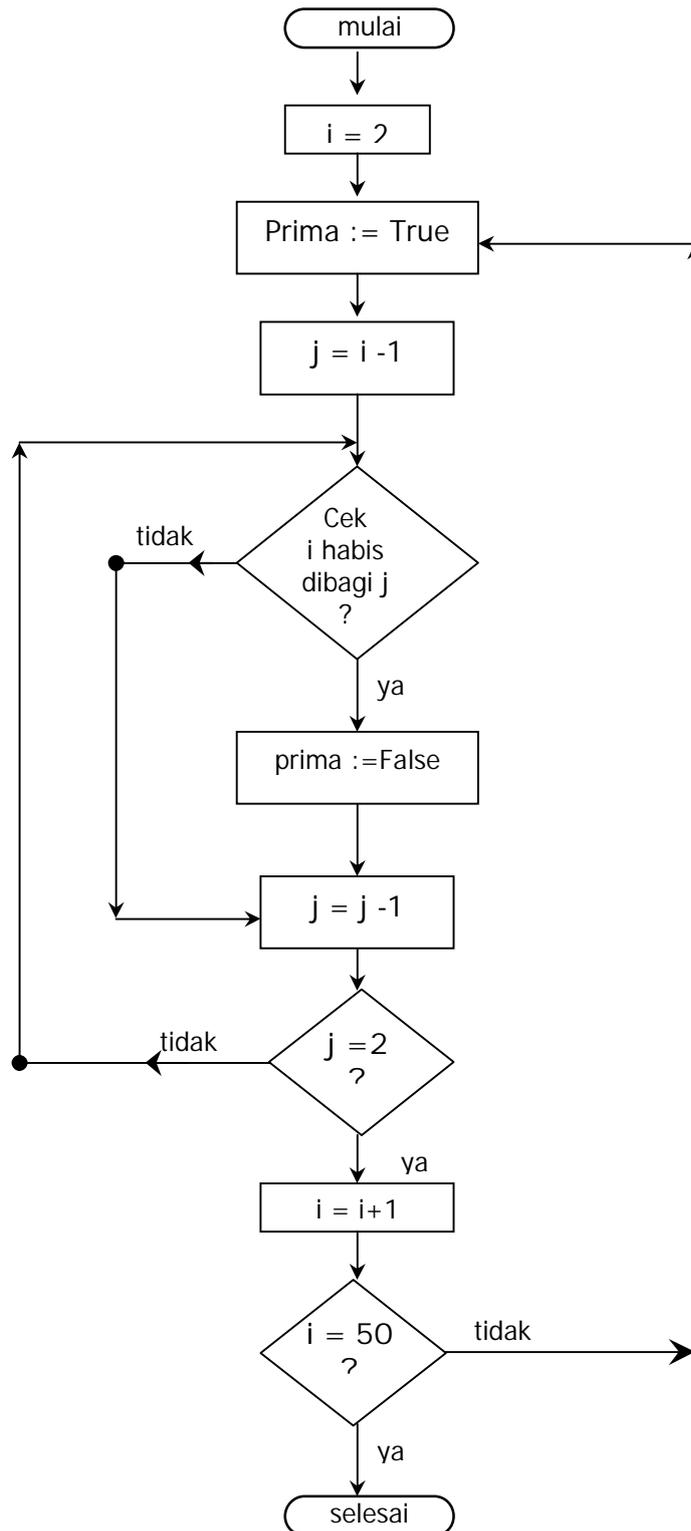
for i:=2 to 50 do
  begin
    prima:=True;
    for j:=i-1 downto 2 do
      if ( (i mod j)=0 ) then prima:=False;
    if (prima) then writeln(i);
  end;

```

keterangan:

mod adalah fungsi untuk menemukan hasil bagi dari suatu operasi pembagian, contoh: $11 \text{ mod } 3$ hasilnya 2.

Diagram alirnya sebagai berikut.



Gambar 12. Diagram alir untuk jawaban soal evaluasi nomor 4

5. Algoritma untuk mencari data maksimum (atau minimum) dalam suatu larik 1 dimensi.

Langkah 1: Mulai

Langkah 2: Tentukan $\text{max} := \text{nilai}[1]$; (asumsikan elemen-1 adalah nilai maksimum).

Langkah 3: Tentukan $i := 2$;

Langkah 4: Cek apakah $\text{nilai}[i] > \text{max}$? Jika tidak, pergi ke langkah 6, Jika ya lanjutkan ke langkah 5 berikut ini.

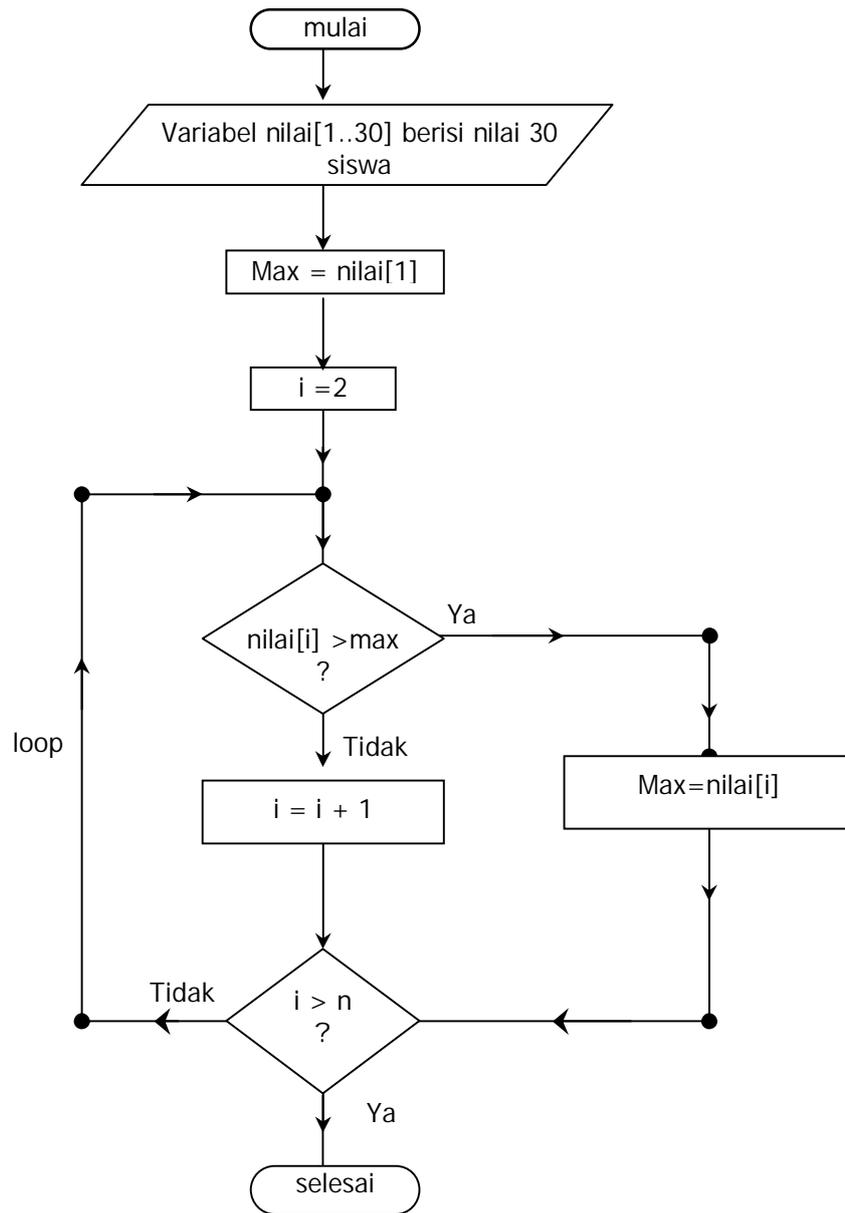
Langkah 5: Ganti isi max dengan $\text{nilai}[i]$, $\text{max} := \text{nilai}[i]$;

Langkah 6: $i := i + 1$; Apakah $i > n$? (n banyaknya elemen larik)
Jika tidak, pergi ke langkah 4, Jika ya pergi ke langkah 7 berikut.

Langkah 7: Nilai maksimum ketemu, yakni max , lalu tampilkan max .

Langkah 8: Selesai.

Diagram alirnya sebagai berikut.



Gambar 13. Diagram alir untuk jawaban soal evaluasi nomor 5

C. KRITERIA KELULUSAN

Aspek	Skor (1-10)	Bobot	Nilai	Keterangan
Kognitif (soal no 1 s/d 3)		4		Syarat lulus nilai minimal 70 dan skor setiap aspek minimal 7
Kebenaran logika dan penggambaran simbol diagram alir		3		
Kerapian, kebersihan, ketelitian		2		
Ketepatan waktu		1		
Nilai Akhir				

Kategori kelulusan:

70 – 79 : Memenuhi kriteria minimal. Dapat bekerja dengan bimbingan.

80 – 89 : Memenuhi kriteria minimal. Dapat bekerja tanpa bimbingan.

90 – 100 : Di atas kriteria minimal. Dapat bekerja tanpa bimbingan.

BAB IV

PENUTUP

Demikianlah modul pembelajaran tentang menyiapkan dan melakukan survey untuk menentukan kebutuhan data. Materi yang telah dibahas dalam modul ini masih sangat sedikit. Hanya sebagai dasar saja bagi peserta diklat untuk belajar lebih lanjut. Diharapkan peserta diklat memanfaatkan modul ini sebagai motivasi untuk menguasai teknik analisis dan design pemrograman lebih jauh sehingga peserta diklat dapat merancang berbagai macam program komputer yang lebih berkualitas.

Setelah menyelesaikan modul ini dan mengerjakan semua tugas serta evaluasi maka berdasarkan kriteria penilaian, peserta diklat dapat dinyatakan lulus/tidak lulus. Apabila dinyatakan lulus maka dapat melanjutkan ke modul berikutnya sesuai dengan alur peta kedudukan modul, sedangkan apabila dinyatakan tidak lulus maka peserta diklat harus mengulang modul ini dan tidak diperkenankan mengambil modul selanjutnya.

DAFTAR PUSTAKA

- Insap Santoso, 1995. Struktur Data Dengan Turbo Pascal, Andi Offset Yogyakarta
- Roger S.Pressman,Ph.D,1997, Rekayasa Perangkat Lunak Pendekatan Praktisi (Buku Satu) Edisi Bahasa Indonesia, Penerbit Andi Offset Yogyakarta